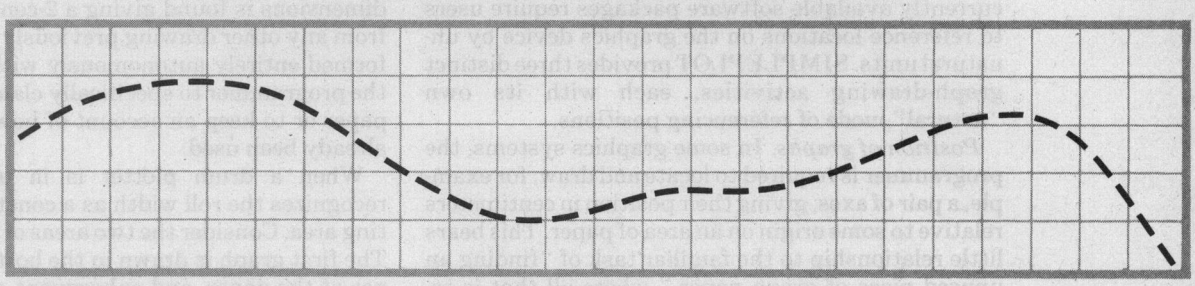
With suitable software, some graphics applications can be brought within reach of general computer users who have specific graphical requirements. One such package is SIMPLEPLOT.



Special Feature:

An Easy-to-Use Graph Drawing Package

Judy Butland and S. D. Butland
University of Bradford

Scientists and engineers whose interest in programming is subsidiary to their main research activities, and who produce theories and results which can be represented graphically, have wasted a great deal of time either plotting manually, or developing often elaborate programs to draw graphs. The plotting software that has been made available has often been too clever, offering so many flexible facilities that the non-expert has difficulty in relating them to his own circumscribed requirements. Or else it has not been clever enough, often providing only a rudimentary set of graphical facilities that produce stylized and unprofessional-looking output. It was this environment that led to the design of SIMPLEPLOT.

Intended for research workers who are not primarily computer programmers but who nevertheless want to produce high-quality graphs for inclusion in reports, theses, and professional journals, SIMPLEPLOT was designed in close collaboration with users to ensure that wherever possible the range of pictures produced relates closely to their actual requirements, and that the instructions necessary to produce their pictures relate to the manual production of graphs with which they are all familiar. Here are some of its facilities:

Framework definitions
- linear and/or logarithmic axes
- Smith charts
- polar axes

Plotting within frameworks
- straight lines from point to point
- smooth curves through sets of points
- curves of functions from function definitions
- circles
- different types of broken lines to distinguish curves
- different types of symbols to mark points

Plotting complete composite graphs
- histograms
- contour graphs
- hidden-line surface drawings

Layout, etc.
- keys
- titles
- layout in pages
- organization within pages

Four general rules have been consistently observed throughout the development of the subroutines:

(1) Do not require the programmer to make decisions which can be made equally well or better by the package.
(2) Minimize use of core store, mill time, and plotter time.
(3) Evade errors, but give intelligible diagnostic information when errors are detected.
(4) Leave all parameters unchanged.

Decision removal

Opening the plotter. The plotter opening or device allocation is performed automatically as soon as the first feasible graph is attempted. This involves deferring the call of the subroutine to open the plotting device until all checks on parameters to the first subroutine purporting to draw a graph have been checked for consistency. Once the device has been opened, a flag held in COMMON is set to indicate

0018-9162/80/0200-0069$00.75 © 1980 IEEE

that plotting has started. The subroutine to open the plotting device is one of the small number of subroutines which form the device interface and which contain all the device dependent code.

Referencing positions on the plotting area. Many currently available software packages require users to reference locations on the graphics device by unnatural units. SIMPLEPLOT provides three distinct graph-drawing activities, each with its own "natural" mode of referencing positions.

Position of graphs. In some graphics systems, the programmer is required to locate and draw, for example, a pair of axes, giving their position in centimeters relative to some origin on an area of paper. This bears little relationship to the familiar task of "finding an unused piece of graph paper," where all that is required is to be presented with a clean sheet of paper of given dimensions. Where the paper is, on this or that part of the desk, is of little significance, as it is "handy." SIMPLEPLOT therefore relieves the programmer from having to know how his graphs are arranged on paper (where this is the output medium). Whenever a new pair of axes or new 3-D plot, etc., is required, an area of unused paper is automatically claimed.

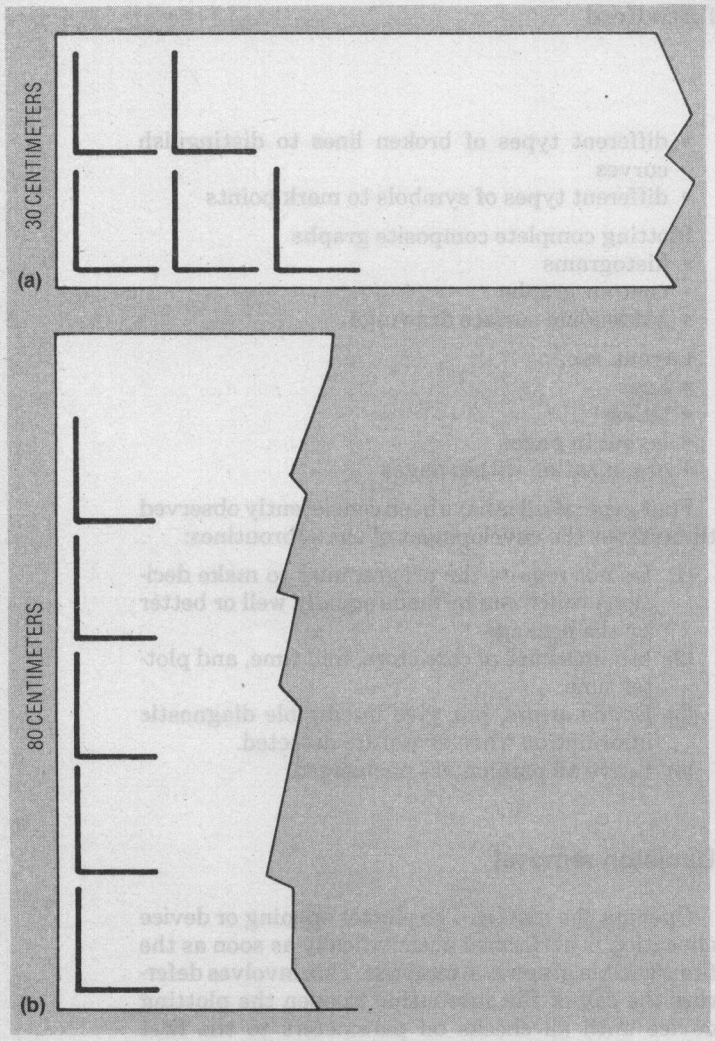


Figure 1. Layout of five graphs, each measuring 12 centimeters by 10 centimeters, where the width of the available paper is (a) 30 centimeters and (b) 80 centimeters.

At the simplest level, an area of paper of given dimensions is found giving a 2-centimeter clearance from any other drawing previously done. This is performed entirely autonomously without the need for the programmer to specifically claim an extra area of paper or to keep an account of how much paper has already been used.

When a drum plotter is in use, the package recognizes the roll width as a constraint on the plotting area. Consider the two areas of paper in Figure 1. The first graph is drawn in the bottom left-hand corner of the paper, and subsequent graphs are drawn above the previous drawing as long as there remains enough room on the paper. When no further room can be found, a new "column" of graphs is drawn clear of the widest graphs in the previous line. A default 2-centimeter clearance is left below and to the left of each graph. Parameters to the system in which SIMPLEPLOT is to be implemented include the default length and width of paper available. These are set up when SIMPLEPLOT is implemented (see Appendix 3). Parameters which can be defined by the programmer include

- alternative width to be used where a two-width plotting device is available at a particular installation,
- the maximum length to be used when the programmer requires to claim more paper than the default length made available, and
- the margin between individual graphs.

The facility also exists to nominate the dimensions of pages on which graphs or sets of graphs are to be drawn. In the simple case, each graph is set within a separate page. Where the dimensions of the graph are sufficiently smaller than the page size, it is set within the page dividing the unused space in proportions 2:1—i.e.,

2 left:1 right;
2 bottom:1 top.

These proportions were chosen after consultation with a number of research workers and after determining the general practice in papers and theses. Where the dimensions of a graph fail to fit within the page specified, the graph is reduced to fit within the page with 3 centimeters to spare for a border, retaining the desired proportions.

In the case where more than one graph is required within a page, the programmer has to work out the total area to be occupied by a set of graphs in one page, and define this area—the "packing area"—before he draws them. The packing area is filled from left to right, from the top down. The 2-centimeter gap between graphs must be included in the calculation of the area required. Thus, six graphs 6 centimeters by 5 centimeters may be packed in a page in two rows of three by defining a packing area—

width = 3*(6+2) = 24 centimeters,
height = 2*(5+2) = 14 centimeters;

or in three rows of two (see Figure 2) by defining a packing area—

width = 2*(6+2) = 16 centimeters,
height = 3*(5+2) = 21 centimeters.

The page size and packing area size only have to be given when they change, and all subsequent plotting is laid out according to the latest details. This usually means that layout is defined at the beginning of a program and not referred to again. If a packing area is defined which is too large for the page size, it is reduced to fit in the page; if the first graph of a page is too large for the packing area, it is reduced to fit in the packing area, and all the graphs in the page are reduced by the same factor so that the required proportions are maintained.

Plotting on graphs. With an area of paper automatically claimed and, say, a pair of axes drawn within the area, the programmer's own units, used by SIMPLEPLOT to mark off the axes, are then used to refer to the coordinates of points for plotting.

Coordinates may be expressed as Cartesian (X,Y) or polar (R, Theta), with Theta expressed in degrees or radians. For a Smith chart, the complex value to be plotted may be passed by the programmer to SIMPLEPLOT, which will then convert this value to appropriate Cartesian coordinates for plotting.

The same plotting subroutines operate on linear or logarithmic axes, polar graphs, or Smith charts; the interpretation of the coordinates is recognized within the subroutines by checking flags set up when each graph is started, but may be redefined during the graph drawing process—e.g., switching from polar in degrees to polar in radians.

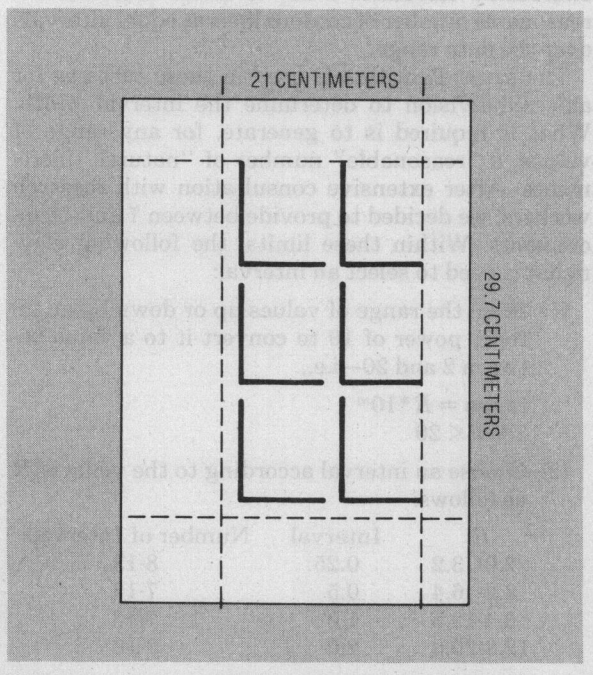
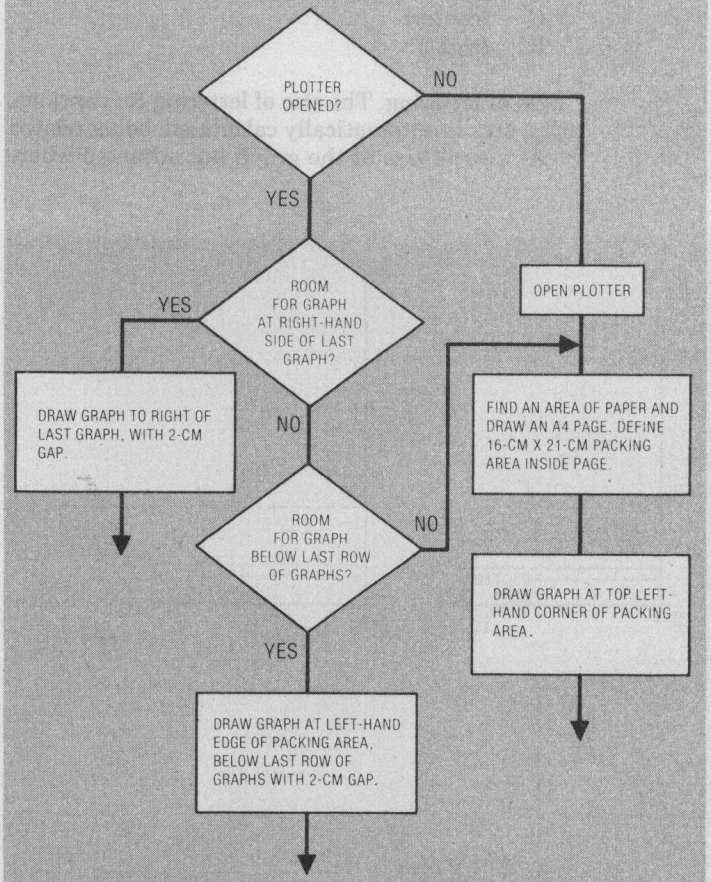


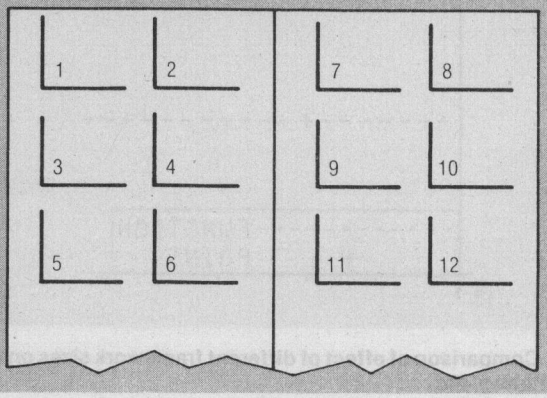
Figure 2. Example layout of graphs using PACKIN.

Example

User program	*Effect hidden inside SIMPLEPLOT*
(1) CALL PAGE (21.0, 29.7)	Note that subsequent plotting must be in A4 pages. Do no plotting.
(2) CALL PACKIN (16.0, 21.0)	Note that subsequent graphs should be packed together in an area 16 centimeters x 21 centimeters. Do no plotting.
(3) Commands to draw a graph 6 centimeters x 5 centimeters	See flow chart below.


PLOTTER OPENED?
NO
YES
ROOM FOR GRAPH AT RIGHT-HAND SIDE OF LAST GRAPH?
YES
OPEN PLOTTER
DRAW GRAPH TO RIGHT OF LAST GRAPH, WITH 2-CM GAP.
NO
FIND AN AREA OF PAPER AND DRAW AN A4 PAGE. DEFINE 16-CM X 21-CM PACKING AREA INSIDE PAGE.
ROOM FOR GRAPH BELOW LAST ROW OF GRAPHS?
NO
DRAW GRAPH AT TOP LEFT-HAND CORNER OF PACKING AREA.
YES
DRAW GRAPH AT LEFT-HAND EDGE OF PACKING AREA, BELOW LAST ROW OF GRAPHS WITH 2-CM GAP.

If (3) is repeated 12 times, the graphs are arranged like this:



Labeling of graphs. Anyone wishing to set a title or key on a graph by hand will tend to think of it in general terms relative to the graph as a whole, and will want the lettering to be a "reasonable" rather than specific size. The position of titles and keys is therefore given in descriptive rather than coordinate form, referring to their general location. Titles and keys may be placed vertically:

'T' (top of graph)
'B' (bottom of graph)
'H' (higher than top)
'L' (lower than bottom)

and horizontally:

'L' (left)
'C' (center)
'R' (right)

Size of lettering. The size of lettering for captions, titles, etc., is automatically calculated, being related to the overall size of the graph but adjusted where necessary to make the required text fit in the length available. The basic character size S is calculated when setting up a graph:

S = (sum of axis lengths)*0.004 centimeters

subject to a minimum of 0.14 centimeters. This equation was arrived at after experiment, taking into account the comments on alternatives by the research workers for whom the software was originally designed. If the character size of the output device being addressed is fixed, then that size is used. The basic character size S relates to the width of character, and the distance between lines of text is calculated on the assumption that no device draws characters of worse proportions than the line printer, with proportions of height to width being 5:3 (including gaps between the lines and characters).

Individual character sizes, etc., are related to S as follows:

titles	S*2.0
symbols	S
captions for keys	S*1.2
"broken line" increments	S*0.7

The effect of these proportions is to ensure that titles are drawn with larger characters than captions, and that the size of symbols plotted and the gaps in broken lines relate to the overall size of the graph. Examples of the effect of thus determining the relative sizes of lettering and "broken line" increments are given in Figure 3.

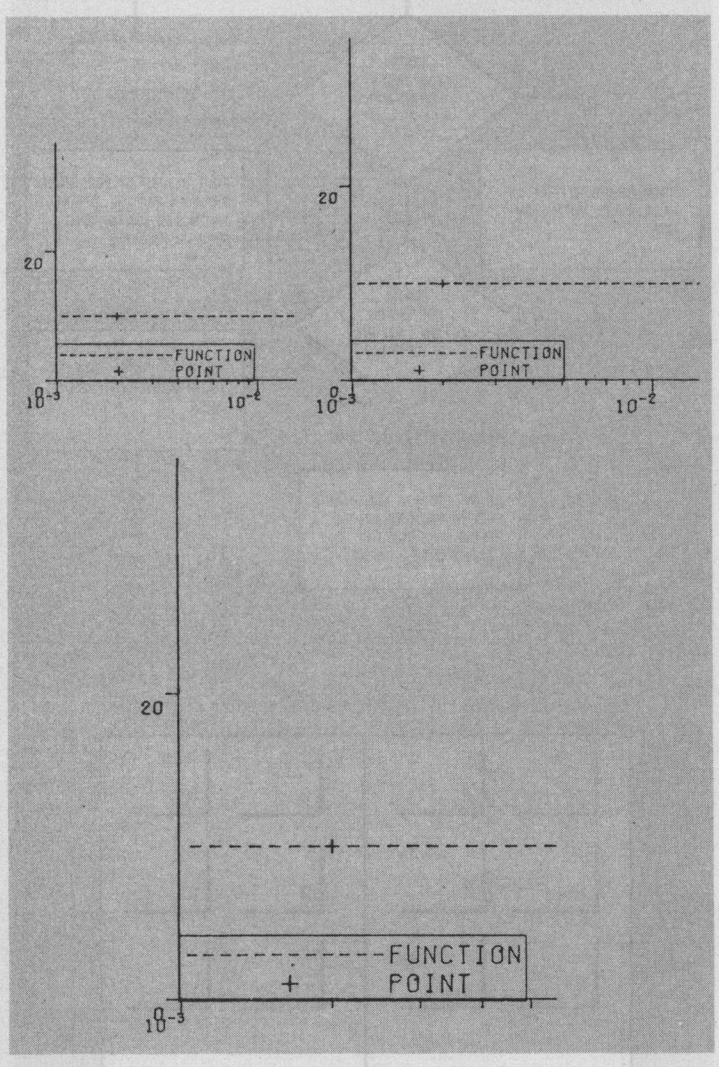


Figure 3. Comparison of effect of different framework sizes on the size of the lettering, etc.

Choice of intervals. The histogram subroutine operates on raw data, counting frequencies of values in a set of equal intervals before drawing the graph; the programmer may either specify the width of intervals himself, or he may leave it to SIMPLEPLOT to make a sensible choice. Similarly, the contour graph subroutine calculates a set of levels and draws a reasonable number of contour lines at equal intervals over the data range.

The same algorithm is used in these cases as for axis subdivision to determine the interval width. What is required is to generate, for any range of values, a "reasonable" number of "natural" increments. After extensive consultation with research workers, we decided to provide between 7 and 13 increments. Within these limits, the following algorithm is used to select an interval:

(1) Scale the range of values up or down by an integer power of 10 to convert it to a value between 2 and 20—i.e.,

$\text{range} = R*10^m$
$2 \leq R < 20$

(2) Choose an interval according to the value of R as follows:

R	Interval	Number of Intervals
2.0- 3.2	0.25	8-13
3.2- 6.4	0.5	7-13
6.4-12.8	1.0	7-13
12.8-20.0	2.0	7-10

(3) Multiply the interval by 10^m.

Labeling axes. To ensure that the increment values on the axes are readable, the following rules are observed:

(1) All values on the increments are restricted to six characters at most.
(2) Whole number increments are drawn as integers (without a decimal point).
(3) Figures which would exceed six characters are scaled, and the power of 10 by which they are to be multiplied is added to the caption on the axis—e.g., 0.00000075 will be represented on the axis as 0.75 and the scale, $*10^{-6}$, will be added to the caption.

The effect of writing different ranges of values on axes of the same length is given in Figure 4.

Provision of defaults. Wherever choices are offered to the programmer, defaults are offered when possible, so that the learner can master a few rudimentary facilities initially and add enhancements to a working program later.

Efficiency

In addition to the obvious attempts to produce compact and efficient code, extra constraints have been self-imposed throughout the development of SIMPLEPLOT.

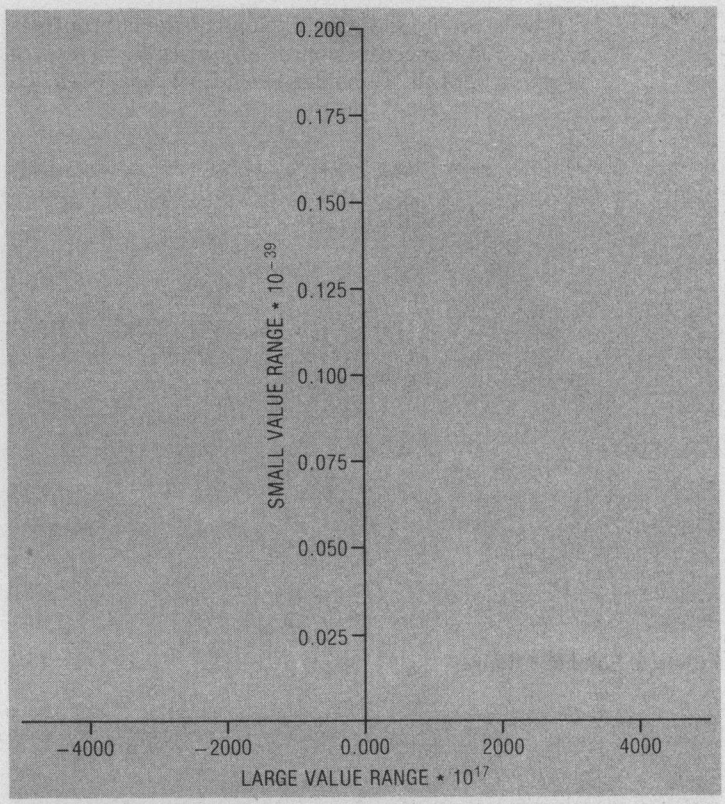


Figure 4. Effect of different ranges on axis lettering.

Video Digitizer 270A

- A TV to Computer Input Device
- A Computer to Film Output Device
- Computer Controllable Sampling Patterns and Rates
- 480 x 512 Element Resolution, 8 Bit Gray Scale

Model 270A lets your computer look at the whole world—economically. Four-page data sheet includes specs, block diagram, flow chart.
Model 270A $4000.

Colorado Video, Inc.
P. O. Box 928 Boulder, CO 80306 (303) 444-3972 TWX 910-940-3248 COLO VIDEO BDR
Video Instruments for Data Acquisition, Processing, Transmission, Display

Reader Service Number 10

Core store minimization. None of the subroutines requires the specification of any working arrays of variable length. Considerable effort has been expended in developing methods for such tasks as drawing smooth curves through sets of points, drawing contour graphs, and drawing hidden-line surfaces which do not require any working arrays and yet produce good quality graphs. Two extra bonuses derive from this principle:

(1) All parameters the programmer is required to specify are of direct relevance to him.
(2) The software is uniquely applicable to implementation on computing systems with restricted core.

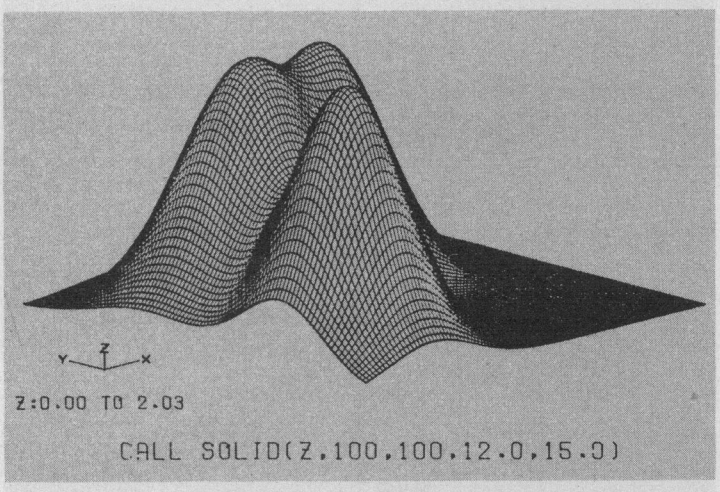


Figure 5. Full solid figure.

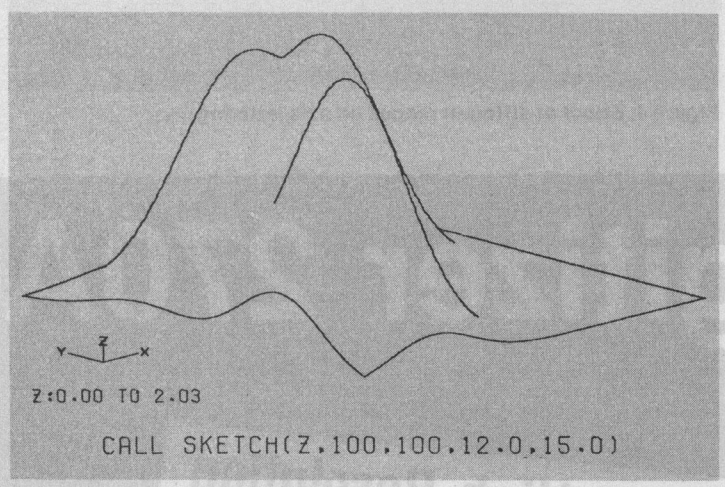


Figure 6. Outline figure of Figure 5.

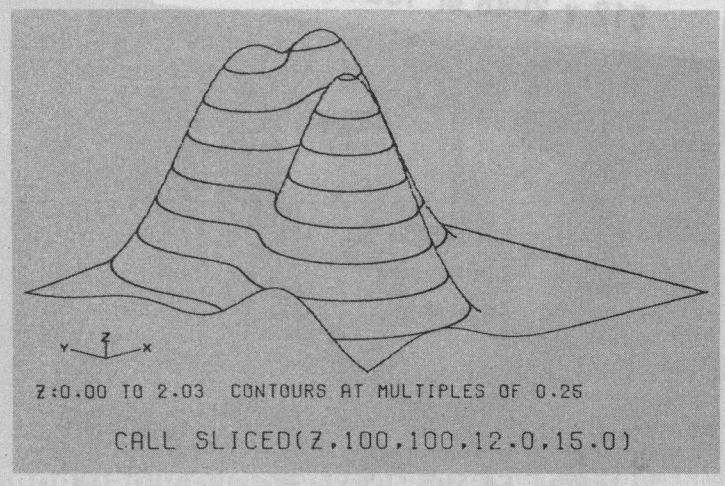


Figure 7. Outline figure with contour markings.

Plotting time minimization. When composite tasks such as drawing axes or 3-D pictures are specified completely in a single subroutine call, the pen movements are arranged to define an efficient traveling distance to achieve the composite drawing. The hidden-line surface picture subroutines use different colors where they are available on an installation in order to distinguish the upper and lower sides of the surface, and all the plotting of one color is completed before the other color is used so that the drawing is not slowed down by repeated pen changes.

A completely new set of subroutines have been made available to draw the outline alone of a "hidden-line" surface, using the same data and parameters as the subroutine to draw the full cross-hatched picture. This offers the advanatges of efficiency in both mill and plotter time, and it gives a good approximation to the function being drawn. See Figures 5-7.

Error actions

All recognizable errors are evaded, and suitable messages are directed to the line printer. After each graph is finished, a report is sent to the line printer indicating whether or not any attempt was made to draw outside the area requested. Attempts to plot outside the limits of the current graph are always ignored. An attempt to draw a smooth curve through two points gives a straight line, though fewer than two are ignored; unrealizable scales such as logarithmic scales through zero, or a scale with equal minimum and maximum values, are omitted, and then all plotting calls are ignored until the next one which sets up a new graph. It has been assumed that a set of incomplete graphs is very useful in debugging a plotting program, and there are therefore no conditions in which identifiable errors in attempting to plot cause the program to halt. A full set of error messages and their significance are given in Appendix 1.

Preserving data

No parameter values are altered within the subroutines. (There is a single exception to this rule: when one is plotting values on a Smith chart, a complex value is supplied to SIMPLEPLOT, which calculates the Cartesian coordinates on the plot corre-

sponding to this complex value. These coordinates are then returned to the user program as parameters.) The programmer using them may therefore perform further calculations on the same data after plotting, and literal parameters are always acceptable for single variables in subroutine calls.

The package has been so designed that its use parallels manual graph drawing, and graph plotting may be added to existing programs without fundamentally altering the program structures.

Structure of programs using SIMPLEPLOT

There are several setting-up subroutines which define the type of graph to be drawn next—e.g., Cartesian graph with linear and/or logarithmic axes, polar graph, Smith chart. Each requires parameters relevant to the requirements of a graph of that type. The first feasible call of a setting-up routine causes tho opening of the plotter, and subsequent ones cause a new unused area of paper to be found or a new picture to be started on a screen. The basic framework of the graph which defines its scales is drawn by the setting-up subroutine. Within such a framework, the subroutines, for joining points by straight lines or curves, or for marking points or for plotting functions may be invoked to plot the graph itself. The same subroutines can be used on linear or logarithmic Cartesian scales or polar graphs—the interpretation of the coordinates is determined by the type of graph being drawn. It is possible to switch between degrees and radians, and between polar Cartesian interpretation of coordinates. Histograms, contour graphs, and hidden-line surface drawings are each produced in their entirety from a single subroutine call which will initiate both the setting-up procedure to find a new area of paper and also the relevant plotting.

Additional labeling in the form of titles and keys can be added easily to a graph at any stage of its production once it has been set up. Service subroutines are included for enabling the request of more paper than a given default (this must be invoked before the plotting device is opened), and for producing graphs on "pages" of a chosen size and with specified gaps between them. These have a continuing effect having been called once, and they apply to all subsequent graphs until another call to change them. The closing of the plotter has to be specifically requested by the user program, since no way could be found to make this automatic using standard Fortran IV.

STATE-OF-THE-ART WORKSTYLE.
STATE-OF-THE-SUNSHINE LIFESTYLE.
ADD IT UP. IT'S YOUR FUTURE!

The state-of-the-art at Harris Controls Division.

It's a half a century of redefining and forwarding the concept of supervisory control and data acquisition (SCADA) systems.

It's current expansion to large scale energy management systems and power control centers around the world.

For you, it's unencumbered opportunity.

Opportunity to be recognized and rewarded in a rapidly progressing field.

Opportunity to design and produce in a professional environment that encourages contribution.

And yes.

Opportunity to live a beautiful, affordable lifestyle on Florida's scenic, Atlantic shoreline.

Add it all up. *It's your future!*

SOFTWARE DESIGN/
SENIOR SOFTWARE DESIGN
ENGINEERS

Design, development and integration background. Five-plus years experience in Operating Systems Development, Device Handlers, Distributed Processing, Computer-to-Computer Communication, Telecommunication, Diagnostic Development and/or Large Scale Integration and Test. Task leadership positions encompass design and project responsibilities and technical supervision.

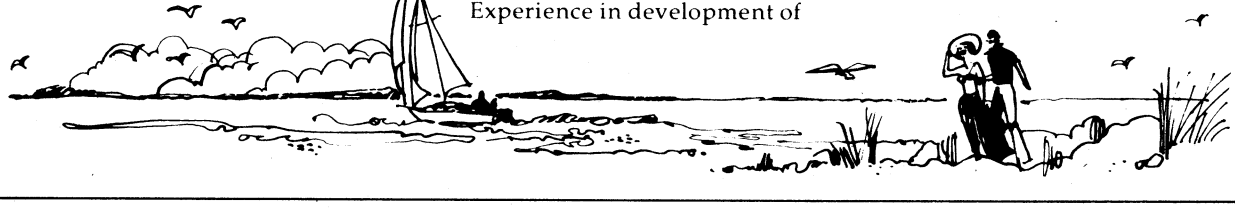
POWER SYSTEM ANALYSTS/
ENGINEERS

Experience in development of control and planning solutions for real-time systems associated with computers in power industry problems. Familiarity with current analytical techniques for the solution of load flow, contingency conditions, load forecasting and state estimation. EE majors with graduate degrees, and AGC/EDC background.

Send resume with salary history to: R.B. Storch, Harris Corporation, Controls Division, P.O. Box 430IEC, Melbourne, Florida 32901.

HARRIS
COMMUNICATION AND INFORMATION PROCESSING

An Equal Opportunity Employer M/F



Implementation of SIMPLEPLOT on different configurations

The subroutines have been designed to be transferrable easily to different devices. The transfer involves setting about twelve common variables and writing a few small subroutines to SIMPLEPLOT specifications. The common variables define system constants such as default and maximum paper sizes, and conversion factor between system units of measurement and centimeters. Each small subroutine performs one simple task such as "move pen," "write text," "open plotter," "close plotter"; when facilities are available on the system, subroutines may be included to "change pen" or "change paper width"; SIMPLEPLOT contains coding for drawing symbols, but if system hardware symbols are preferred, a "draw symbol" subroutine may be supplied. Successful implementations have already been carried out on a wide range of mainframes, from PDP-11 to CDC 7600, driving a variety of devices including drum plotters, flat-bed plotters, and display devices.

SIMPLEPLOT is designed as a special-purpose graphics package for drawing graphs and does not attempt to serve as a general-purpose plotting package. Within the applications covered by SIMPLEPLOT, generality has deliberately been sacrificed for ease of use whenever we thought these were in conflict. Graphics standards, therefore, simplify the interfacing of SIMPLEPLOT to different systems while in no way altering the user interface. It has already been interfaced to widely differing graphical devices and systems through their own basic plotting software including PLOT10, Calcomp, GINO-F, GHOST, SPECTRE, and many low-level graphics systems. We expect any form of standard which might be adopted to represent a suitable vehicle for SIMPLEPLOT.

Comparison of SIMPLEPLOT with other packages

SIMPLEPLOT was originally developed to satisfy local requirements when the alternative software available at the university was that supplied by Calcomp with their plotters. Every facility offered was developed with ease of use as the prime consideration, and all routine tasks which are performed by the majority of graph drawing programs were absorbed into the package to relieve the user of the burden of organization. It appears that SIMPLEPLOT has taken the concept of user orientation further than other packages for drawing graphs, particularly in the organization of layout and label positioning. The effect of this on the task of writing a program to plot graphs is best illustrated by comparing the programming of a simple graph drawing problem using SIMPLEPLOT and other packages.

A test example. For illustration, a simple plotting example has been chosen which requires a set of commonly needed features together from one program. Programs have been written to use the Calcomp software, GINOGRAF (a graph-drawing extension to the GINO-F package), and SIMPLEPLOT to make full use of the facilities offered in each case to draw the specified graphs from simple programs (see Figures 8-13). The chosen example is as follows:

(1) Read one set of 11 abscissae and two sets of 11 corresponding ordinates.
(2) Draw two separate graphs 12 centimeters by 6 centimeters, one above the other.
(3) On the top graph, plot the two sets of points by marking them by two distinct symbols.

```
C DIMENSION ARRAYS TO ALLOW EXTRA ELEMENTS FOR
C FIRSTV & DELTAV VALUES, AND FOR ZERO TO INCLUDE
C IN SCALE
      DIMENSION X(14),Y(13,2)
      READ(1,1)(X(I+1),Y(I,1),Y(I,2),I=1,11)
1     FORMAT(F5.1,F6.3,F5.2)
C SET ZERO VALUES IN ARRAY ELEMENTS NOT CONTAINING
C DATA
      X(1)=0.0
      Y(12,1)=0.0
      Y(13,1)=0.0
C WORK OUT SCALES FOR BOTH AXES
      CALL SCALE(X(1),12.0,12,1)
      CALL SCALE(Y(1,1),6.0,24,1)
C COPY SCALING PARAMETERS FOR Y TO END OF FIRST
C DATA SET
      Y(12,1)=Y(12,2)
      Y(13,1)=Y(13,2)
C START PLOTTING
      CALL PLOTS(0,0,7)
C MOVE ORIGIN TO SUITABLE POSITION FOR FIRST GRAPH
      CALL PLOT(7.0,19.0,-3)
C CALL OWN SUBROUTINE TO DRAW AXES
      CALL SETUP(X(13),X(14),Y(12,1),Y(13,1),7HSYMBOLS,7)
C PLOT EACH SET OF DATA BY MARKING WITH SYMBOLS
      CALL LINE(X(2),Y(1,1),11,1,-1,1)
      CALL LINE(X(2),Y(1,2),11,1,-1,2)
C MOVE ORIGIN TO SUITABLE POSITION FOR SECOND GRAPH
      CALL PLOT(0.0,-8.0,-3)
C CALL OWN SUBROUTINE TO DRAW AXES
      CALL SETUP(X(13),(X(14),Y(12,1),Y(13,1),5HLINES,5)
C PLOT EACH SET OF DATA BY JOINING WITH STRAIGHT LINES
      CALL LINE(X(2),Y(1,1),11,1,0,0)
      CALL LINE(X(2),Y(1,2),11,1,0,0)
C WRITE TITLE IN SUITABLE POSITION
      CALL SYMBOL(0.0,-5.0,0.42,7HCALCOMP,0.0,7)
C CLOSE PLOTTER
      CALL PLOT(0.0,0.0,999)
      STOP
      END


C DRAW A PAIR OF AXES USING XFIRST,XDELTA TO SCALE
C HORIZ. AXIS,AND YFIRST,YDELTA TO SCALE VERT. AXIS
C WRITE NCAP CHARACTERS FROM CAP AS A TITLE
      SUBROUTINE SETUP(XFIRST,XDELTA,YFIRST,YDELTA,
     1 CAP,NCAP)
      DIMENSION CAP(1)
C DRAW AXES
      CALL AXIS(0,0,0,0,8HABSCISSA,-8,12.0,0.0,XFIRST,
     1 XDELTA)
      CALL AXIS(0.0,0.0,8HORDINATE,8,6.0,90.0,YFIRST,
     1 YDELTA)
C WRITE TITLE AT A SUITABLE PLACE ON THE GRAPH
      CALL SYMBOL(0.1,6.0,0.35,CAP(1),0.0,NCAP)
      RETURN
      END
```

Figure 8. Fortran listing of Calcomp test program.

```
      DIMENSION X(11),Y(11,2)
      READ(1,1)(X(I),Y(I,1),Y(I,2),I=1,11)
1     FORMAT(F5.1,F6.3,F5.2)
C SCAN DATA TO FIND RANGE OF VALUES
C START WITH ZEROS SO THAT ZERO INCLUDED IN SCALES
      XMIN=0.0
      XMAX=0.0
      YMIN=0.0
      YMAX=0.0
      DO 10 I=1,11
      IF(X(I).LT.XMIN)XMIN=X(I)
      IF(X(I).GT.XMAX)XMAX=X(I)
      DO 10 J=1,2
      IF(Y(I,J).LT.YMIN)YMIN=Y(I,J)
      IF(Y(I,J).GT.YMAX)YMAX=Y(I,J)
10    CONTINUE
C START PLOTTING
      CALL CN925
C DEFINE PAGE SIZE
      CALL DEVPAP(210.0,290.0,0)
C FIX CHARACTER SIZE
      CALL CHASIZ(3.0,3.0)
C FIX SCALES OF AXES
      CALL AXISCA(1,8,XMIN,XMAX,1)
      CALL AXISCA(1,8,YMIN,YMAX,2)
C CALL OWN SUBROUTINE TO DRAW AXES IN SUITABLE POSITION
      CALL SETUP(70.0,190.0,9HSYMBOLS*.)
C PLOT EACH SET OF DATA BY MARKING WITH SYMBOLS
      CALL GRASYM(X,Y(1,1),11,7,0)
      CALL GRASYM(X,Y(1,2),11,1,0)
C CALL OWN SUBROUTINE TO DRAW AXES IN SUITABLE POSITION
      CALL SETUP(70.0, 110.0,15HSMOOTH CURVES*.)
C PLOT EACH SET OF DATA BY JOINING WITH SMOOTH CURVES
      CALL GRACUR(X,Y(1,1),11)
C SWITCH TO BROKEN LINES FOR SECOND DATA SET
      CALL DASHED(-1,2.0,1.0,0.0)
      CALL GRACUR(X,Y(1,2),11)
C MOVE PEN TO SUITABLE PLACE TO WRITE TITLE
      CALL MOVT02(70.0,50.0)
      CALL CHAHOL(10HGINOGRAF*.)
C CLOSE PLOTTER
      CALL DEVEND
      STOP
      END



C DRAW A PAIR OF AXES STARTING AT (XLOC,YLOC)
C WRITE A TITLE FROM CAP
      SUBROUTINE SETUP(XLOC,YLOC,CAP)
      DIMENSION CAP(1)
C DEFINE POSITION OF GRAPH
      CALL AXIPOS(0,XLOC,YLOC,120.0,2)
      CALL AXIPOS(0,XLOC,YLOC,60.0,1)
C DRAW HORIZ. AXIS AND WRITE LABEL ON IT
      CALL AXIDRA(2,1,1)
      CALL MOVT02(XLOC+40.0,YLOC-10.0)
      CALL CHAHOL(10HABSCISSA*.)
C DRAW VERT. AXIS AND WRITE LABEL ON IT
      CALL AXIDRA (-2,-1,2)
      CALL MOVT02(XLOC-10.0,YLOC+20.0)
C CHANGE ANGLE OF TEXT WRITING
      CALL CHAANG(90.0)
      CALL CHAHOL(10HORDINATE*.)
C RESTORE HORIZONTAL TEXT WRITING
      CALL CHAANG(0.0)
C MOVE PEN TO SUITABLE PLACE,AND WRITE TITLE
      CALL MOVT02(XLOC+1.0,YLOC+50.0)
      CALL CHAHOL(CAP)
      RETURN
      END
```

Figure 9. Fortran listing of GINOGRAF test program.

(4) On the bottom graph, plot the two sets of points by joining them by two separate curves.
(5) For both graphs, draw axes scaled to suit the data, but including zero in both scales.
(6) Write captions on each axis of each graph, a title on each graph, and a title for the page containing the graphs.

The programs have been written assuming no knowledge of the data values so that the plotting is related directly to values which are read, including the fixing of scales. The data values used were these:

X	*Y*1	*Y*2
20	0.040	0.02
40	0.042	0.02
60	0.045	0.02
80	0.048	0.02
100	0.052	0.02
120	0.280	0.02
140	0.124	0.03
160	0.088	0.04
180	0.076	0.06
200	0.068	0.12
220	0.064	0.28

```
C DIMENSION ARRAYS TO ALLOW EXTRA ELEMENTS FOR ZEROS
C TO INCLUDE IN SCALES
      DIMENSION X(12),Y(12,2)
C SPECIFY PAGE SIZE AND SUITABLE LAYOUT
      CALL PAGE(21.0,29.7)
      CALL PACKIN(14.0,16.0)
      READ(1,1)(X(I),Y(I,1),Y(I,2),I=1,11)
1     FORMAT(F5.1,F6.3,F5.2)
C SET ZERO VALUES IN ARRAYS FOR INCLUSION IN SCALING
      X(12)=0.0
      Y(12,1)=0.0
C DRAW AXES FOR FIRST GRAPH
      CALL JBAXES(X,12,12.0,8HABSCISSA,8,Y,23,6.0,
     1 8HORDINATE,8)
C WRITE TITLE AT TOP RIGHT
      CALL TITLE(1HT,1HR,7HSYMBOLS,7)
C PLOT EACH SET BY MARKING WITH SYMBOLS
      DO 10 J=1,2
      DO 10 I=1,11
10    CALL MARKPT(X(I),Y(I,J),J)
C DRAW KEY TO SYMBOLS AT TOP LEFT
      CALL SET KY(1HT,1HL,2,15)
      CALL MARK KY(1,14HFIRST DATA SET,14)
      CALL MARK KY(2,15HSECOND DATA SET,15)
C DRAW AXES FOR SECOND GRAPH
      CALL JBAXES(X,12,12.0,8HABSCISSA,8,Y,23,6.0,
     1 8HORDINATE,8)
C WRITE TITLE AT TOP RIGHT
      CALL TITLE(1HT,1HR,13HSMOOTH CURVES,13)
C PLOT EACH SET OF DATA BY JOINING WITH SMOOTH CURVES
      CALL BRKN CV(X,Y(1,1),11,0)
      CALL BRKN CV(X,Y(1,2),11,-1)
C DRAW KEY TO LINES AT TOP LEFT
      CALL SET KY(1HT,1HL,2,15)
      CALL LINE KY(0,14HFIRST DATA SET,14)
      CALL LINE KY(-1,15HSECOND DATA SET,15)
C WRITE PAGE TITLE IN CENTER,LOWER THAN GRAPHS
      CALL TITLE(1HL,1HC,10HSIMPLEPLOT,10)
C CLOSE PLOTTER
      CALL ENDPLT
      STOP
      END
```

Figure 10. Fortran listing of SIMPLEPLOT test program.

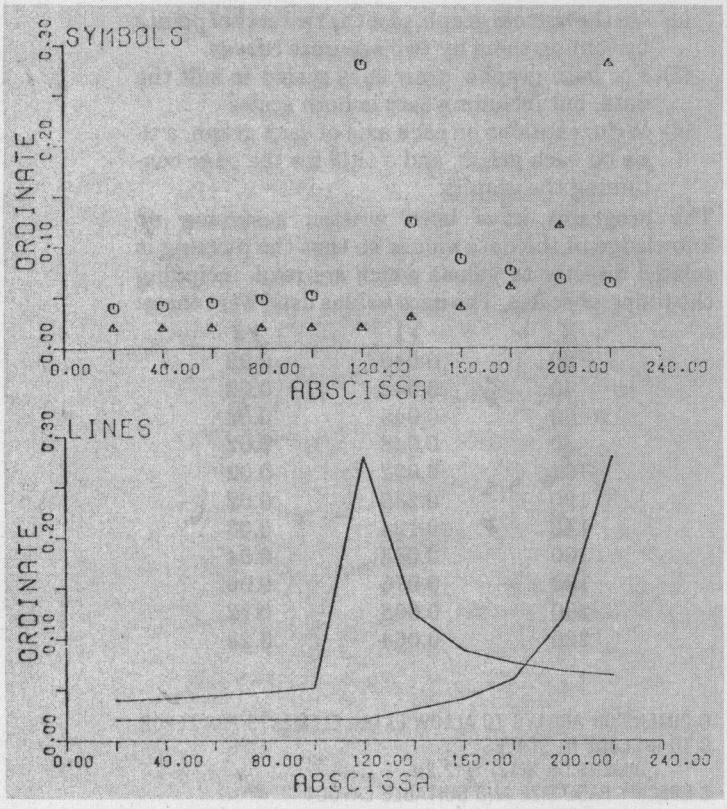


Figure 11. Plotter output from Calcomp test program.

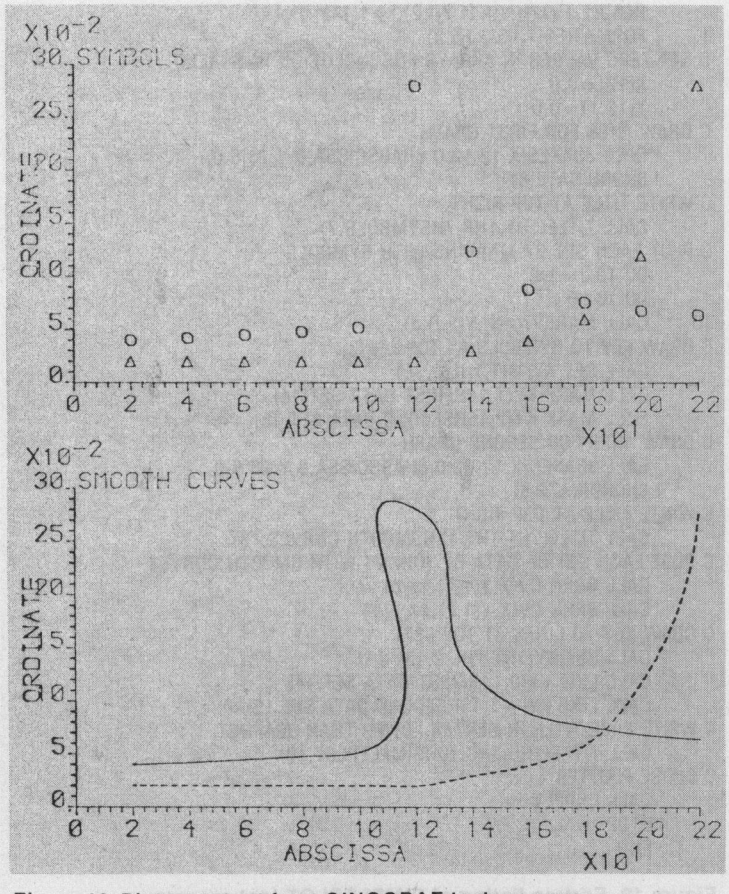


Figure 12. Plotter output from GINOGRAF test program.

Comparison of test programs. Comments have been included throughout the test program listings (Figures 8-10) so that those unfamiliar with the particular packages should have no difficulty in understanding each line. Apart from the obvious differences in methods required for programming the different parts of the example, it should be noted how often the word "suitable" occurs in the comments in the Calcomp and GINOGRAF programs. Every time the comment refers to " a suitable position," the program contains numerical values which have had to be calculated in advance by the programmer, usually by drawing a sketch of the required layout. The single use of the word "suitable" in the SIMPLEPLOT comments relates to the preliminary layout statement; everything afterwards is positioned in an appropriate place by the package, including keys which are not available as such in the other packages.

Because SIMPLEPLOT is designed specifically for graph drawing rather than general plotting, some of the flexibility offered by other graphics packages has been strategically taken away from the user on the assumption that this flexibility complicates the plotting of graphs. The major gain over other packages is believed to be achieved in two general areas:

(1) Three types of locations in the plotting area are treated in different ways from each other, but as the non-progammer would expect:

- Positions of graphs are either automatically allocated or allocated in terms of the dimensions of the areas they occupy.
- Labeling on graphs is positioned descriptively.
- Plotting on graphs is described in units of the graph scales.

The programmer is never required to use any coordinate system except for the plotting on his graph.

(2) The provision of labeling facilities allows captions and keys to be written and related to existing plotting very easily, and all lettering is written in sizes set by the package as appropriate to the function of the text and also the area available for the label.

Appendix 1. Monitoring of graphs; error actions

Whenever possible, SIMPLEPLOT takes evasive action to avoid failure of plotting programs containing errors. To assist interpretation of results, messages are displayed on monitoring files to show whether or not errors have been avoided, and also to show at which stage the plotter is opened

and closed. One message is generated for every attempted graph, one message is generated if the plotter is opened, and one message is generated if the plotter is closed. If a program terminates for any reason before the plotter is closed, the message relating to the latest graph may be absent. The messages coming out of SIMPLEPLOT are enclosed in round brackets ( ) to distinguish them from other displays on the monitoring file.

MESSAGE	*COMMENTS*
(PLOTTER OPENED BY SIMPLEPLOT)	
(PLOTTER CLOSED BY SIMPLEPLOT)	
(GRAPH COMPLETE)	No errors detected
(GRAPH INCOMPLETE)	Some out-of-range plotting omitted
(GRAPH OMITTED)	Plotter already closed by program
(GRAPH OMITTED:TOO BIG)	Complete graph omitted because of errors in parameters used in setting up of graph
(GRAPH OMITTED:CONSTANT DATA)	
(GRAPH OMITTED:ZERO LENGTH)	
(GRAPH OMITTED:ZERO COUNT)	
(GRAPH OMITTED:LOG AXIS THROUGH ZERO	
(SURFACE OMITTED:CONSTANT DATA)	
(SURFACE OMITTED:ZERO LENGTH)	
(SURFACE OMITTED:ZERO COUNT)	
(CONTOUR GRAPH OMITTED:CONSTANT DATA	
(CONTOUR GRAPH OMITTED:ZERO COUNT)	
(HISTOGRAM OMITTED:ZERO COUNT)	

Errors in the number or types of parameters are not detectable from within the subroutines, and therefore cause undefined failures.

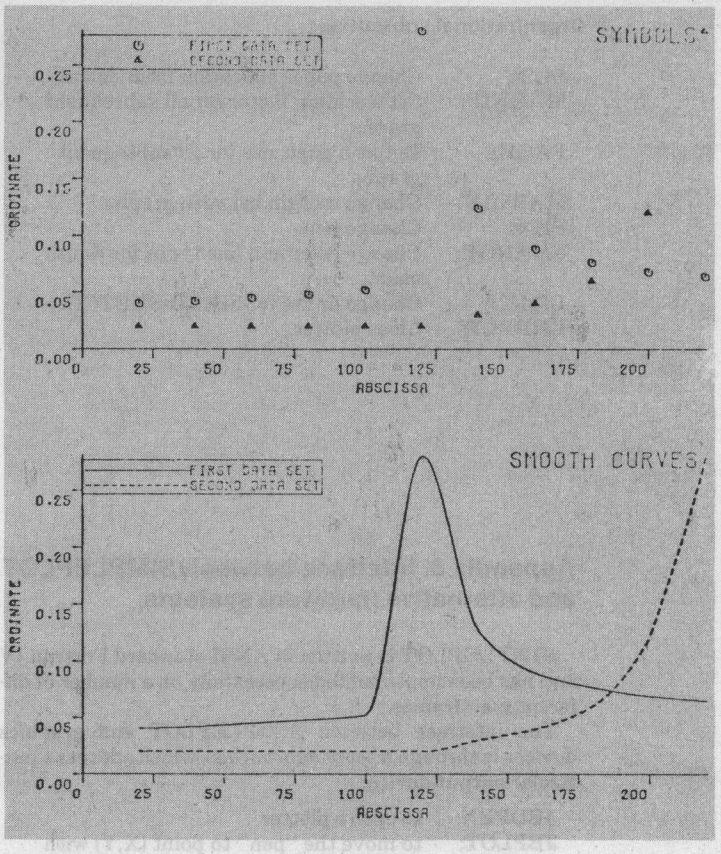


Figure 13. Plotter output from SIMPLEPLOT test program.

Appendix 2. Subroutines offered in SIMPLEPLOT

Subroutines for special graphs

HISTGM:	Draw histogram.
SOLID:	Draw cross-hatched hidden-line surface picture.
SKETCH:	Draw outline hidden-line surface picture.
SLICED:	Draw a sketch with contours on it.
CONTR:	Draw contour graph.

Setting-up subroutines

NEWPLT:	Set up Cartesian graph with no drawing.
JBAXES:	Set up Cartesian graph and draw axes.
JBPOLE:	Set up polar graph and draw axis and circle.
SMITH:	Set up Smith chart.

Graph plotting subroutines

JOINPT:	Join this point to last with straight line.
BREAK:	Lift pen (between two successions of JOINPT).
MARKPT:	Mark point with a selected symbol.
DRAWLN:	Draw line between point and zero.
XRANGE:	Mark a range of X values.
YRANGE:	Mark a range of Y values.
DRAWCV:	Draw a smooth curve through a set of points.
BRKNCV:	Draw a smooth broken curve through a set of points.
DRAWFN:	Draw a smooth curve from a function definition.
BRKNFN:	Draw a smooth broken curve from a function definition.
FNRNGE:	Change range for DRAWFN/BRKNFN from default.
DRAWCL:	Draw a circle.
BRKNCL:	Draw a broken circle.
JBCODE:	Calculate Smith chart coordinates from complex value.
CARTES:	Switch coordinate interpretation to Cartesian.
DEGS:	Switch coordinate interpretation to polar in degrees.
RADS:	Switch coordinate interpretation to polar in radians.

Graph labeling subroutines

DRAW AX:	Draw a single axis.
TITLE:	Write a title on graph.
SET KY:	Initialize a key.
LINE KY:	Write a key for a broken line.
MARK KY:	Write a key for a symbol.
BOTH KY:	Write a key for a line and symbol together.
BLNK KY:	Include extra space between key entries.

Organizational subroutines

SIZE:	Change paper allocation from default.
MAGNIF:	Set a scaling factor for all subsequent graphs.
PAGE:	Define a page size for all subsequent graphs.
MARGIN:	Change margin between graphs.
PEN:	Change pen.
CHANGE:	Change color and line types for Smith chart.
LIM3D:	Change limits for SOLID/SKETCH.
ENDPLT:	Close plotter.

Appendix 3. Interface between SIMPLEPLOT and alternative hardware systems

SIMPLEPLOT is written in ANSI standard Fortran IV and has been implemented successfully on a number of different mainframes.

The interface between SIMPLEPLOT and graphics devices is through a set of subroutines which address a particular output device:

JBOPEN:	to open a plotter
JBPLOT:	to move the "pen" to point (X,Y) with X,Y in system units with or without drawing
JBTEXT:	to write text
JBEND:	to close a plotter
JBDISP:	output a message of 40 characters to the monitoring device (e.g., line printer)
JBPEN:	to select a pen (if more than one is available)
JBNEW:	to set up a new picture
JBSYMB:	to draw a special symbol
JBCHNG:	to alter width of paper to be used from default
JBDATA:	to assign values to COMMON variables

Default settings relating to a particular device are set up in a COMMON area and are accessed by the above interface subroutines. They indicate

- whether special symbols need to be coded by SIMPLEPLOT,
- whether text may be rotated,
- whether the character width is variable,
- the conversion factor from centimeters to system units,
- default paper size (width and length),
- whether either default paper length or width may be changed by program,
- whether a graph may be turned through 90° on the output device,
- maximum size of an individual graph, and
- whether the size of an individual graph is fixed.

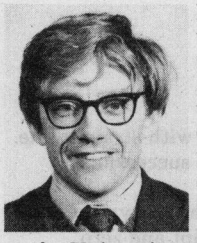
Setting up the above subroutines has proved to be a trivial task for all new output devices on which SIMPLEPLOT has been made available so far; as new devices become available, we expect the provision of the additional interface to be correspondingly simple. ■

References

1. J. Butland, "Surface Drawing Made Simple," *Computer-Aided Design*, Vol. 11, No. 1, Jan. 1979, pp. 19-22.
2. *Calcomp Reference Manual*, Calcomp, Fullerton, Calif., 1969.
3. *GINOGRAF User Manual*, CAD Centre, Cambridge, UK, 1976.
4. *SIMPLEPLOT User's Handbook*, University of Bradford, UK, 1978.

Judy Butland has been computer officer for the Postgraduate School of Electrical and Electronic Engineering at the University of Bradford since 1970. She writes programs, assists research students in programming, and collaborates in the solution of computational problems associated with a wide variety of research projects in electrical engineering. It is in this environment that her research interests in providing simple graphic facilities for engineers have developed.

She studied mathematics at the University of Manchester Institute of Science and Technology from 1960 to 1963, but was first introduced to computing when working as a mathematical assistant for Manchester Business School in the University of Manchester in 1967. She received an M. Phil at University of Bradford in 1979 for her work on SIMPLEPLOT.



David Butland is the head of User Services in the Computing Laboratory at the University of Bradford, England. He is primarily concerned with the provision of central computing facilities required by research projects at the University. His own research interests center around the generation of test sequences for large digital networks.

Butland received a BA in philosophy at the University of Manchester in 1964 and a PhD at the University of Bradford in 1978 for his work on the generation of diagnostic tests. He is a member of the British Computer Society and is married to Judy Butland.